

Appl. No. : 09/826,607
Filed : April 5, 2001

AMENDMENTS TO THE CLAIMS

Please amend Claims 1, 14-16, 18, 21, 22, 26-29, 31, 33-35, and 39 as indicated below.

Please add new Claims 40-52 as indicated below.

1. (Currently Amended) A process for converting a conventionally coded computer application program into a data set suitable for streamed delivery across a network from a server ~~and concurrent execution on~~ to a client in a computer environment, comprising the steps of:

providing installation monitoring means for monitoring an ~~the~~ installation process of said conventionally coded application program on a local computer system;

wherein said installation monitoring means monitors ~~the system registry~~ modifications that said installation process makes to certain file paths in a the system registry of said local computer system ~~and records the system modification data~~;

~~wherein said installation monitoring means monitors and records any file modifications made by said installation process;~~

~~sorting said system modification data and said file modification data and removing duplicate entries;~~

parameterizing said system registry modifications by replacing certain of said file paths in said system registry modifications with parameters that are recognizable by said client ~~all of said local computer system's specific registry keys, value names, and values in said system modification data and said file modification data; and~~

providing data set creation means for processing said parameterized system registry ~~modifications~~ data ~~and said parameterized file modification data~~ to create a data set suitable for streaming said parameterized system registry modifications over said network such that said application program is capable of beginning execution on said client prior to downloading all of said application program.

Appl. No. : **09/826,607**
Filed : **April 5, 2001**

2. (Original) The process of claim 1, wherein said data set creation means creates a runtime data set, said runtime data set consists of all regular application files and directories containing information about said regular application files.

3. (Original) The process of claim 2, wherein said data set creation means creates an initialization data set that is the first set of data streamed from said server to said client, said initialization data set prepares said client for streaming of said runtime data set.

4. (Original) The process of claim 2, wherein said directories contain lists of file names, file numbers, and the metadata associated with the files in a particular directory.

5. (Original) The process of claim 1, wherein said data set creation means creates a versioning table that contains a list of root file numbers and version numbers for tracking application patches and upgrades, and wherein each entry in said versioning table corresponds to one patch level of an application with a corresponding new root directory.

6. (Original) The process of claim 5, wherein said versioning table is sent to said client by said server, said client compares said versioning table with said client's root file number for the particular application program to find the necessary files required for a software upgrade or patch.

7. (Original) The process of claim 1, further comprising the step of: providing a user interface that allows an operator to examine all changes made to said local computer system during said installation process and to edit said system modification data and said file modification data.

8. (Original) The process of claim 1, wherein said installation monitoring means monitors said application program as it runs and is being configured for a particular working environment on said local computer system and records common configurations of said application program thereby allowing said common configurations to be automatically duplicated on other client machines.

9. (Currently Amended) The process of claim 1, further comprising ~~the step of: program profiling~~ means for capturing a the sequence of file blocks being accessed during normal execution of said application program.

Appl. No. : **09/826,607**
Filed : **April 5, 2001**

10. (Original) The process of claim 9, wherein said sequence of file blocks is used to pre-cache frequently used blocks on said client before said application program is first used by a user.

11. (Original) The process of claim 9, wherein said sequence of file blocks is used to optimize large directories of files on said client for faster file accesses.

12. (Original) The process of claim 9, wherein said sequence of file blocks is tied to specific user input and wherein said client pre-fetches file blocks based on user input to said application program.

13. (Original) The process of claim 1, wherein said installation monitoring means records the state of said local computer system before said installation process begins to give a more accurate picture of any modifications that are observed by said installation monitoring means.

Appl. No. : 09/826,607
Filed : April 5, 2001

14. (Currently Amended) An apparatus for converting a conventionally coded computer application program into a data set suitable for streamed delivery across a network from a server ~~and concurrent execution on to a~~ client in a computer environment, comprising:

installation monitoring ~~module that monitors means for monitoring the~~ installation process of said conventionally coded application program on a local computer system;

wherein said installation monitoring ~~module that means monitors the~~ system registry modifications that said installation process makes to certain file paths in the system registry of said local computer system ~~and records the system modification data;~~

~~wherein said installation monitoring means monitors and records any file modifications made by said installation process;~~

~~a module for sorting said system modification data and said file modification data and removing duplicate entries;~~

a module for ~~parameterizing that parameterizes said system registry~~ modifications by replacing certain of said file paths with parameters that are recognizable by said client ~~all of said local computer system's specific registry keys, value names, and values in said system modification data and said file modification data; and~~

data set creation ~~means for processing module that processes said~~ parameterized system registry modifications ~~data and said parameterized file modification data~~ to create a data set suitable for streaming said parameterized system registry modifications over said network such that said application program is capable of beginning execution on said client prior to downloading all of said application program.

Appl. No. : 09/826,607
Filed : April 5, 2001

15. (Currently Amended) The apparatus of claim 14, wherein said data set creation ~~module means~~ creates a runtime data set, said runtime data set consists of all regular application files and directories containing information about said regular application files.

16. (Currently Amended) The apparatus of claim 15, wherein said data set creation ~~means~~ module creates an initialization data set that is the first set of data streamed from said server to said client, said initialization data set prepares said client for streaming of said runtime data set.

17. (Original) The apparatus of claim 15, wherein said directories contain lists of file names, file numbers, and the metadata associated with the files in a particular directory.

18. (Currently Amended) The apparatus of claim 14, wherein said data set creation ~~means~~ module creates a versioning table that contains a list of root file numbers and version numbers for tracking application patches and upgrades, and wherein each entry in said versioning table corresponds to one patch level of an application with a corresponding new root directory.

19. (Original) The apparatus of claim 18, wherein said versioning table is sent to said client by said server, said client compares said versioning table with said client's root file number for the particular application program to find the necessary files required for a software upgrade or patch.

20. (Original) The apparatus of claim 14, further comprising: a user interface that allows an operator to examine all changes made to said local computer system during said installation process and to edit said system modification data and said file modification data.

21. (Currently Amended) The apparatus of claim 14, wherein said installation monitoring ~~means~~ module monitors said application program as it runs and is being configured for a particular working environment on said local computer system and records common configurations of said application program thereby allowing said common configurations to be automatically duplicated on other client machines.

22. (Currently Amended) The apparatus of claim 14, further comprising: program profiling ~~means for capturing~~ module that captures the sequence of file blocks being accessed during normal execution of said application program.

Appl. No. : **09/826,607**
Filed : **April 5, 2001**

23. (Original) The apparatus of claim 22, wherein said sequence of file blocks is used to pre-cache frequently used blocks on said client before said application program is first used by a user.

24. (Original) The apparatus of claim 22, wherein said sequence of file blocks is used to optimize large directories of files on said client for faster file accesses.

25. (Original) The apparatus of claim 22, wherein said sequence of file blocks is tied to specific user input and wherein said client pre-fetches file blocks based on user input to said application program.

26. (Currently Amended) The apparatus of claim 14, wherein said installation monitoring ~~means~~ module records the state of said local computer system before said installation process begins to give a more accurate picture of any modifications that are observed by said installation monitoring ~~means~~.

Appl. No. : 09/826,607
Filed : April 5, 2001

27. (Currently Amended) A program storage medium readable by a computer, tangibly embodying a program of instructions executable by the computer to perform method steps for converting a conventionally coded computer application program into a data set suitable for streamed delivery across a network from a server and concurrent execution on to a client in a computer environment, comprising the steps of:

~~providing installation monitoring means for monitoring the installation process of said conventionally coded application program on a local computer system;~~

~~wherein said installation monitoring means monitors monitoring the system registry modifications that said installation process makes to certain file paths in the system registry of said local computer system and records the system modification data;~~

~~wherein said installation monitoring means monitors and records any file modifications made by said installation process;~~

~~sorting said system modification data and said file modification data and removing duplicate entries;~~

~~parameterizing said system registry modifications by replacing certain of said file paths with parameters that are recognizable by said client all of said local computer system's specific registry keys, value names, and values in said system modification data and said file modification data; and~~

~~providing data set creation means for processing said parameterized system registry modifications data and said parameterized file modification data to create a data set suitable for streaming over said network such that said application program is capable of beginning execution on said client prior to downloading all of said application program.~~

Appl. No. : **09/826,607**
Filed : **April 5, 2001**

28. (Currently Amended) The method of claim 27, further comprising creating ~~wherein said data set creation means creates~~ a runtime data set, said runtime data set ~~consists of all comprises~~ regular application files and directories containing information about said regular application files.

29. (Currently Amended) The method of claim 28, further comprising creating ~~wherein said data set creation means creates~~ an initialization data set that is the first set of data streamed from said server to said client, said initialization data set prepares said client for streaming of said runtime data set.

30. (Original) The method of claim 28, wherein said directories contain lists of file names, file numbers, and the metadata associated with the files in a particular directory.

31. (Currently Amended) The method of claim 27, wherein further comprising creating ~~said data set creation means creates~~ a versioning table that contains a list of root file numbers and version numbers for tracking application patches and upgrades, and wherein each entry in said versioning table corresponds to one patch level of an application with a corresponding new root directory.

32. (Original) The method of claim 31, wherein said versioning table is sent to said client by said server, said client compares said versioning table with said client's root file number for the particular application program to find the necessary files required for a software upgrade or patch.

33. (Currently Amended) The method of claim 27, further comprising ~~the step of:~~ providing a user interface that allows an operator to examine ~~all~~ changes made to said local computer system during said installation process and to edit said system modification data and said file modification data.

34. (Currently Amended) The method of claim 27, further comprising monitoring ~~wherein said installation monitoring means monitors~~ said application program as it runs and is being configured for a particular working environment on said local computer system and records common configurations of said application program thereby allowing said common configurations to be automatically duplicated on other client machines.

35. (Currently Amended) The method of claim 27, further comprising ~~the step of: program profiling means for capturing a~~ the sequence of file blocks being accessed during normal execution of said application program.

36. (Original) The method of claim 35, wherein said sequence of file blocks is used to pre-cache frequently used blocks on said client before said application program is first used by a user.

37. (Original) The method of claim 35, wherein said sequence of file blocks is used to optimize large directories of files on said client for faster file accesses.

38. (Original) The method of claim 35, wherein said sequence of file blocks is tied to specific user input and wherein said client pre-fetches file blocks based on user input to said application program.

39. (Currently Amended) The method of claim 27, further comprising recording wherein ~~said installation monitoring means records~~ the state of said local computer system before said installation process begins ~~to give a more accurate picture of any modifications that are observed by said installation monitoring means.~~

40. (New) A method for converting an application program into a data set suitable for streamed delivery across a network from a server to a client in a computer environment, the method comprising:

monitoring an installation process of an application program on a local computer system;

monitoring system registry modifications that said installation process makes to certain file paths in a system registry of said local computer system;

parameterizing said system registry modifications by replacing certain of said file paths in said system registry modifications with parameters that are recognizable by a client; and

processing said parameterized system registry modifications to create a data set suitable for streaming said parameterized system registry modifications over said network such that said application program is capable of beginning execution on said client prior to downloading all of said application program.

Appl. No. : **09/826,607**
Filed : **April 5, 2001**

41. (New) The method of claim 40, further comprising creating a runtime data set, said runtime data set consists of regular application files and directories containing information about said regular application files.

42. (New) The method of claim 41, further comprising creating an initialization data set that is the first set of data streamed from said server to said client, said initialization data set prepares said client for streaming of said runtime data set.

43. (New) The method of claim 41, wherein said directories contain lists of file names, file numbers, and the metadata associated with the files in a particular directory.

44. (New) The method of claim 40, further comprising creating a versioning table that contains a list of root file numbers and version numbers for tracking application patches and upgrades, and wherein each entry in said versioning table corresponds to one patch level of an application with a corresponding new root directory.

45. (New) The method of claim 44, wherein said versioning table is sent to said client by said server, said client compares said versioning table with said client's root file number for the particular application program to find the necessary files required for a software upgrade or patch.

46. (New) The method of claim 40, further comprising providing a user interface that allows an operator to examine all changes made to said local computer system during said installation process and to edit said system modification data and said file modification data.

47. (New) The method of claim 40, further comprising monitoring said application program as it runs and is being configured for a particular working environment on said local computer system and records common configurations of said application program thereby allowing said common configurations to be automatically duplicated on other client machines.

48. (New) The method of claim 40, further comprising capturing a sequence of file blocks being accessed during normal execution of said application program.

49. (New) The method of claim 48, wherein said sequence of file blocks is used to pre-cache frequently used blocks on said client before said application program is first used by a user.

Appl. No. : **09/826,607**
Filed : **April 5, 2001**

50. (New) The method of claim 48, wherein said sequence of file blocks is used to optimize large directories of files on said client for faster file accesses.

51. (New) The method of claim 48, wherein said sequence of file blocks is tied to specific user input and wherein said client pre-fetches file blocks based on user input to said application program.

52. (New) The method of claim 40, further comprising recording the state of said local computer system before said installation process begins.

Appl. No. : 09/826,607
Filed : April 5, 2001

SUMMARY OF INTERVIEW

Identification of Claims Discussed

Claim 1 was discussed.

Identification of Prior Art Discussed

U.S. Patent No. 6,457,076 to Cheng et al. and U.S. Patent No. 6,574,618 to Eylon et al. were discussed.

Proposed Amendments

Applicant proposed clarifications to the parameterizing limitation in Claim 1. The Application further proposed clarifying streaming applications.

Principal Arguments and Other Matters

Applicant identified that none of the cited references appeared to disclose the concept of parameterizing certain file path certain file path modifications made in a registry. This parameterized data is then used to create a data set suitable for streaming over a network such that streamed portions of the application are capable of executing on a remote computer prior to the entire application being streamed over the network.

Results of Interview

It was Applicant's understanding that the Examiner agreed that the cited references failed to disclosed the claim limitations and that the Examiner would conduct an additional search.